SPECTRUM TURTLE GRAPHICS

by

Eric Deeson and Keith Jammer

A guide to the AVC program
'turtle'
for the Sinclair ZX Spectrum micro

AVC SOFTWARE

PO BOX 415
HARBORNE
BIRMINGHAM 617 9TT

Revised Edition October 1982

"turtle" and "turtle 2"

These notes apply to low resolution program "turtle". If you have “turtle 2”, please refer also to the insert sheet.

If you have "turtle" you may obtain "turtle 2" on cassette with the insert sheet for £2.50 (overseas £3.00), a reduction of £1.00 on the standard price.

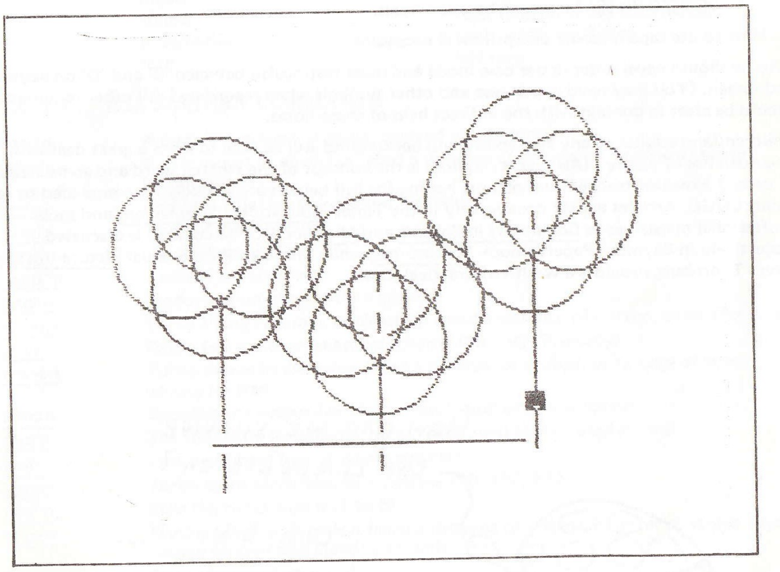
SPECTRUM TURTLE GRAPHICS

by

Eric Deeson and Keith Jammer

A guide to the AVC program
'turtle'
for the Sinclair ZX Spectrum micro

CONTENTS



NOTE

INTRODUCTION

AVC's 'turtle' for the 16K ZX Spectrum, although fairly versatile, does not claim to be a full implementation of LOGO. In being less than complete, however, it is in a number of ways better constructed than true LOGO and is very easy to use in a wide variety of contexts.

There are two main ways in which this version is below par.

1. It generally uses low resolution graphics. This is partly due to the Spectrum's limitations on mixing PRINT and PLOT, and partly due to the need to keep it of a length suitable for the 16K micro.
2. It may not be fully crash-proof, though we hope to have made it so. Again memory restrictions are relevant.

We are, however, working on a more sophisticated version for the 48K machine.

With all their learning software, it is AVC's policy to make the material as user-friendly as possible. LOGO and 'turtle' are systems rather than casual programs, however, and need teacher guidance for most effective development. This booklet is a guide for the teacher or parent rather than for the young explorer.

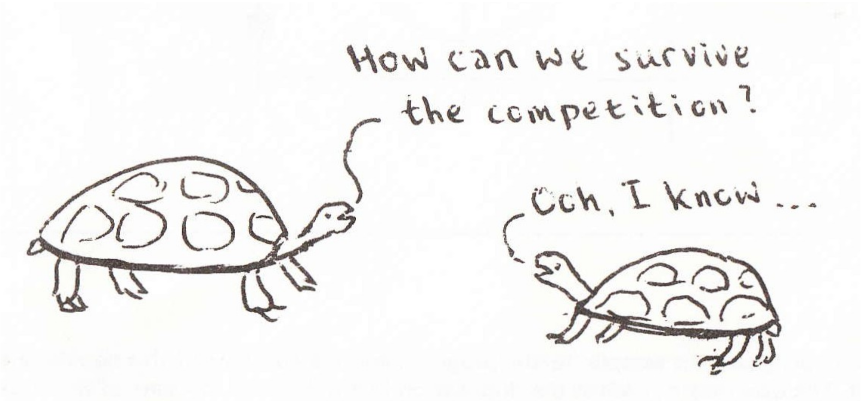
AVC's user-friendliness policy also implies that the actual user should have only minimal knowledge of computers and computing. He or she will need only to know the following:

1. How to use DELETE to correct entries prompted by "L".
2. To use ENTER (E) after such responses.
3. How to obtain decimal point and the symbol '*' used with procedures.
4. How to react to a Spectrum crash if obtained -

   GO TO 50 (E) for a complete re-start;
   GO TO 70 (E) for return to main menu.
5. How to use tape-recorder and printer if necessary.

Also he should never enter upper case mode and must distinguish between '0' and 'O' on keyboard and screen. (You may need upper case and other symbols when recording.) All other requirements should be clear in context with the indirect help of these notes.

Children (and adults) of any age, ability and background will be able to learn a great deal with unguided use of 'turtle'. (All that is required is the concept of the written word and some sense of order.) However teachers and parents hoping for full benefit are strongly recommended to read about LOGO. Articles appear occasionally in the Times' Educational Supplement and such professional magazines as Computers in Schools and Microscope. The concept is discussed discursively in Seymour Papert's book Mindstorms, while the Atari Pilot manual Picture this! by David Thornburg provides a wealth of practical ideas.



THE STRUCTURE OF THE SYSTEM

This version of turtle graphics LOGO offers a main menu with ten options, and 'turtle' routines with these features:

- up to sixty statements ('lines").
- up to five procedures ("sub-routines").
- loops.
- in all, sixteen statement types.

Turtle routines may be edited at any stage, there being special error messages as well as the Sinclair ones (which should be rare). At any stage, too, the current routine may be saved on cassette and/or printed out, in the latter case with a copy of the screen display it produces.

The syntax of 'turtle' is fairly straightforward. Almost all statements consist of a four-letter command word in lower case followed by up to three decimal digits.

The following tables list the commands and summarise their syntax; more detailed notes appear later. Numbers ('n') refer to the low resolution 22 x 32 screen where applicable.

TABLE 1 THE NINE CLASSES OF COMMAND

Feature	*Command(s)*
colour	penc
control	rept use*
draw, low resolution	draw wipe
draw, high resolution	bend
label	*
move	draw jump move wipe
sound	hoot (and all active commands)
termination	endp endr end* stop
turn	left turn

TABLE 2 THE SIXTEEN COMMANDS

bendn A circle of radius n is drawn centred on the turtle

drawn Turtle moves in direction faced a distance of n steps, or to edge of screen, leaving a trail

endx End of 'turtle' routine section, namely

  endp end of main routine

  endr end of repeat loop

  end*n end of procedure n

hoot n Produces a set of n random notes

jumpn Turtle ('frog'?) jumps in direction faced a distance of n steps, or to edge of screen

left n Turtle turns n degrees anti-clockwise (90, 180, 270 only)

moven Turtle moves in direction faced a distance of n steps, or to edge of screen, leaving no trail

pencn Specifies the colour (n) of the next 'bend' or 'draw' action

rept n The following statements, up to 'endr', will be executed n times

stop Last significant line of whole program

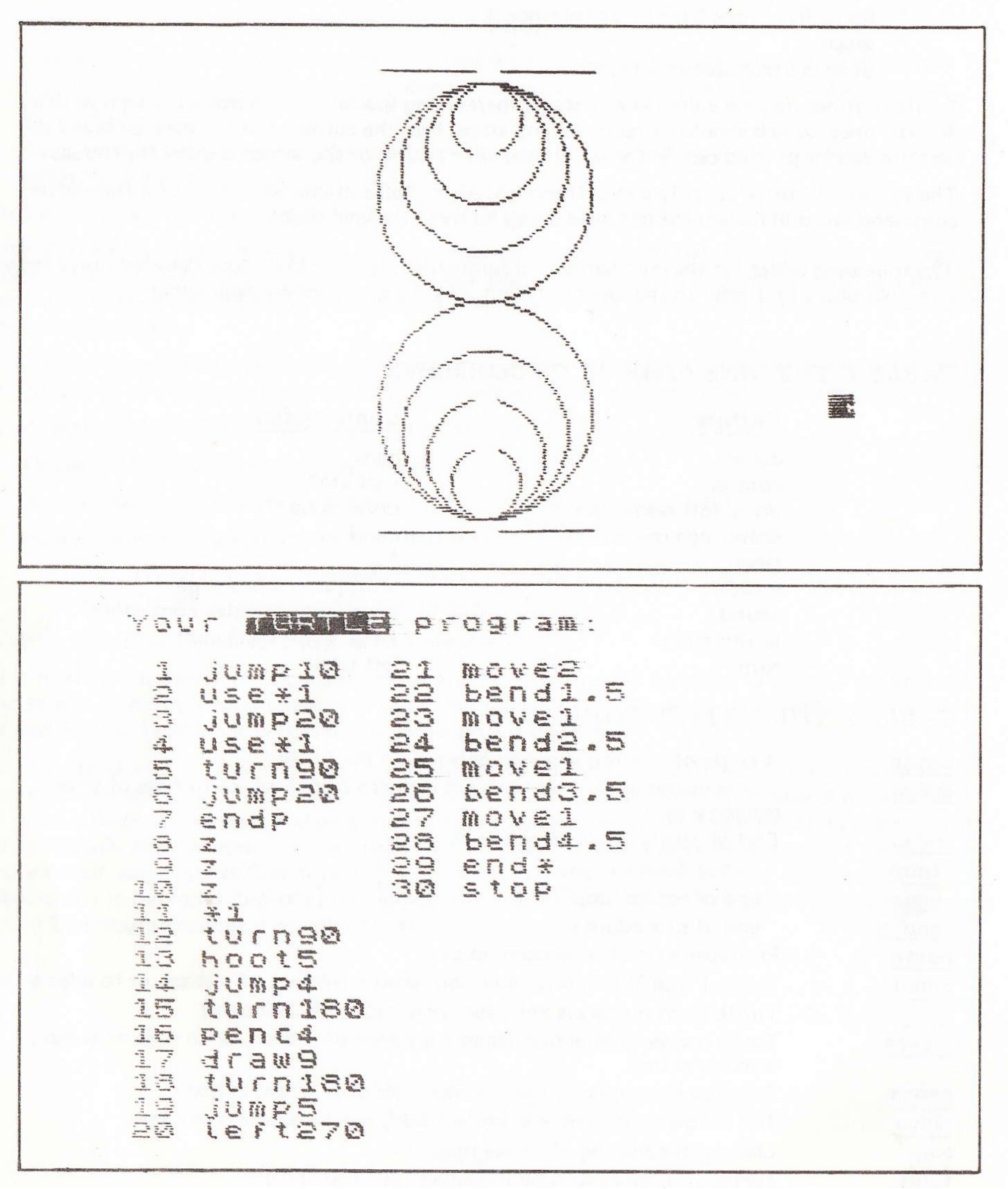
turn n Turtle turns clockwise by n degrees (90, 180, 270)

use*n Calls the procedure n (1 to 5)

wipen Turtle moves in direction faced a distance of n steps, or as far as screen edge, leaving no trail and clearing its path

*n Start of procedure n

The following screen display was obtained using the 'turtle' routine listed below it and option 8 on the main menu. You may also care to consider the demonstration program on the cassette; see later for details about this.



```
Your TURTLE program:

  1 jump10    21 move2
  2 use*1     22 bend1.5
  3 jump20    23 move1
  4 use*1     24 bend2.5
  5 turn90    25 move1
  6 jump20    26 bend3.5
  7 endp      27 move1
  8 z         28 bend4.5
  9 z         29 end*
 10 z         30 stop
 11 *1
 12 turn90
 13 hoot5
 14 jump4
 15 turn180
 16 penc4
 17 draw9
 18 turn180
 19 jump5
 20 left270
```

THE MAIN MENU

This appears

(a) when 'turtle' is first loaded (following a two-second delay);

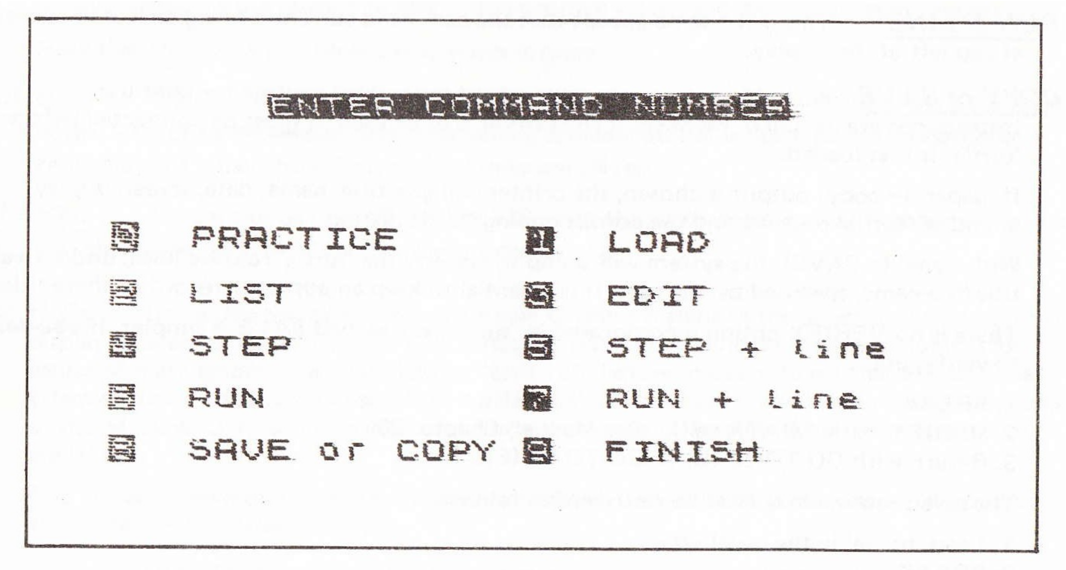
(b) following ENTER after any occasion when 'stop' is encountered;

(c) following ENTER after a 'turtle' error report, except when in practice mode;

(d) following ENTER after 'finish' is entered from the menu;

(e) following GO TO 50 (E) or GO TO 70 (E) after BREAK or a crash.

It looks like this:



```
              ENTER COMMAND NUMBER

   0  PRACTICE          1  LOAD
   2  LIST              3  EDIT
   4  STEP              5  STEP + line
   6  RUN               7  RUN + line
   8  SAVE or COPY      9  FINISH
```

Select the option desired by pressing the corresponding number key. ENTER is not required.

PRACTICE This is the immediate mode (mode 0). Starting at the centre of the screen, the turtle will follow each valid entered statement, printing that statement at the top of the screen. An invalid command leads to an error message and/or it will be ignored. Entries may be corrected and each requires (E) for execution.

It is not meaningful to enter the following while in mode 0: '*', 'reptn' or any 'end'. Entry of 'use*n' will work if that procedure already resides in 'turtle' memory. The entry of 'stop' (or 's') will lead to exit from mode 0, and return to menu follows a further (E).

Instruction sets may be tested at any time in immediate mode. However its use will displace your first line; restore it with 'edit'.

LOAD This allows the line-by-line entry of 'turtle' routines. Line numbers are presented automatically; inputs are confined to seven characters but are not checked other than for 'stop'.

Because there is no re-number facility, experienced users follow each 'end' with a few dummy lines to allow later insertion of extra commands. When 'stop' is met, line entry ceases, and (E) causes return to menu. Note that use of the LOAD option clears the 'turtle' memory.

LIST The list option gives a line-by-line display of the 'turtle' routine currently in memory, up to and including the first 'stop'. Commands after the first 'stop' (if there is more than one) stay in memory but are not displayed. To return to the menu, press (E).

EDIT The current lines are presented one by one with in each case a message requiring a 'y' or 'n' response (with (E)). If 'n' is given, the new command may then be entered

As usual, 'stop', whether met and agreed or newly entered, will end this procedure.

STEP Using this, you may step line by line and action by action through the current 'turtle' routine, until 'stop' is met. (E) initiates each next step and also returns one to the menu at the end.

STEP & LINE This is the same except that the command currently being executed appears at top left of the display.

RUN The complete current routine is executed, from line 1 to 'stop' or 'endp'. Increase execution speed by holding down the ENTER key.

RUN & LINE This is the same except that the command currently being executed appears at top left of the display.

COPY or SAVE With this option, one may record the current routine for later use. Instructions are included. However if the printer is to be used, it must be connected before 'turtle' is first loaded.

If 'paper' (= copy) output is chosen, the printer will give title, name, date, screen display at end of 'turtle' routine, and the corresponding 'turtle' listing.

With 'tape' (= SAVE), the system will dump to cassette the 'turtle' routine lines, under a valid title/file-name, specified by the user. It is essential to keep an adequate record of these titles.

There is no VERIFY option incorporated in 'turtle' as a second SAVE is simpler. If you wish to verify, use:

1. BREAK
2. VERIFY "title" DATA p$(). See Manual, Chapter 20.
3. Return with GO TO 50 (E) or GO TO 70 (E).

The saved routine may later be recovered as follows:

1. Load 'turtle' in the usual way.
2. BREAK.
3. Use LOAD "title" DATA p$(), followed by GO TO 50 or GO TO 70. Any 'turtle' routine in memory already will be lost.

FINISH Use this when you wish to stop and allow someone else to work on their own material. (E) after it returns one to the main menu with loss of stored routine.

THE COMMANDS (remember to use lower case throughout)

BENDn A circle of radius n is drawn centred on the turtle's current position. Any value of n (up to three characters) is allowed — negative values have the same effect as positive ones;

0 has no effect; decimal fractions are accepted. As any attempt to plot outside the display area leads to a BASIC error, values of n that would do this are ignored.

The colour of the circle is blue unless 'bend n' is preceded by a 'penc n' command, with n ≠ 1.

DRAWn The turtle moves in the direction faced (up, right, down or left) a distance of n steps, or to the screen edge, leaving a straight line trail. The trail will erase anything met to a distance of half a character width. Negative values of n are ignored, as is 0; decimal ones are INTed.

The colour of the trail is blue unless the command is preceded by 'pencn' with n ≠ 1.

ENDX This denotes the end of a 'turtle' routine section, as follows:

endp End of main routine ('e' will suffice)

endr End of repeat loop (see REPT)

end* End of procedure (n, 1 to 5, is optional but advisable)

When 'endp' is met, the turtle will halt as if it had met 'stop'. Use this instead of 'stop' when there are any procedures, otherwise these will not appear in the listings.

'endr' is equivalent to BASIC 'next', causing no visible change but affecting control.

'end*' is equivalent to BASIC 'return' or 'endproc' at the end of a closed sub-routine or procedure. It too gives no visible effect.

HOOTn The 'hoot' command causes the turtle to produce a set of n notes, randomly selected within a range that depends on progress through the routine. Here n should be a whole number in the range 1 to 999 — non-positive values and decimal parts are ignored.

Note that the turtle also bleeps as it moves around.

JUMPn This causes the turtle to jump a distance of n units in the direction faced (up, right, down, left) or to the screen edge. Intervening screen material is not affected. Negative and zero values of n have no effect; decimal ones are INTed.

LEFTn Here n can be only 90, 180 or 270, with the turtle rotating that number of degrees anti-clockwise. Use 'turn' for clockwise movement.

MOVEn Using this one commands the turtle to move n steps in the direction faced, or to the display area edge. This version of LOGO has no 'back' command, and negative (and zero) values of n are ignored. Use 'left 180' or 'turn 180' before 'moven' to get that effect, (The effect of 'back' sometimes arises if the turtle is asked to move forward when close to the edge of the window. Only sophisticated users should attempt to employ this.) Decimal parts of n are INTed.

The 'moven' command causes a one-unit wide blank trail to be left; use 'jump n' if you don't want intervening material erased.

PENCn Short for 'pen colour n', this controls the 'ink' used in the next 'bend' or 'draw' action. Valid values of n are 0 to 7 giving the colours marked by those keys. Note that n = 5 gives a trace in background colour, which will not be visible.

TURNn As 'left n' but giving clockwise rotation; n can be only 90, 180 or 270.

REPTn (= repeat n times) This sets up a loop action, executing the material between it and 'endr' n times. Here n should be a positive whole number between 1 and 999. Values below 1 give an error report; decimal parts are ignored. A loop can be of any length (up to 58 lines) but cannot include a second loop. To get that nesting effect, use procedures (see USE*).

STOP (Sophisticated operators may prefer to use 's' alone.) This is used to denote the last line of the complete 'turtle' routine. As such it can appear instead of 'endp', but this is not advisable except where there are no procedures. The only occasion when you may be happy to omit this command is at the end of a sixty-line routine — such a program has an assumed final 'stop'.

Not only does 'stop' prevent error reports in an otherwise correct 'turtle' routine, but it signals the routine's end during a number of menu operations. As in BASIC, 'stop' also has use during development.

*USE*n* With this, one calls for procedure n to be executed. Here n may take only whole number values from 1 to 5. A value of n outside this range gives an error report, while decimal values and those referring to non-existent procedures are ignored. When the procedure reaches 'end', control will return to the next command after 'use*' unless procedure nesting has been attempted. See the entry for *n for more details of 'turtle' procedures.

WIPEn in effect this is exactly the same as 'moven' (and has the same syntax). However its name explicitly indicates its erasing function.

**n* With n taking any whole number value from 1 to 5, this denotes the first line of a procedure (closed sub-routine), equivalent to the BASIC 'def proc'. When the procedure is called, with 'use*n', control passes to the next line after *n and continues to 'end' or 'end* n'.

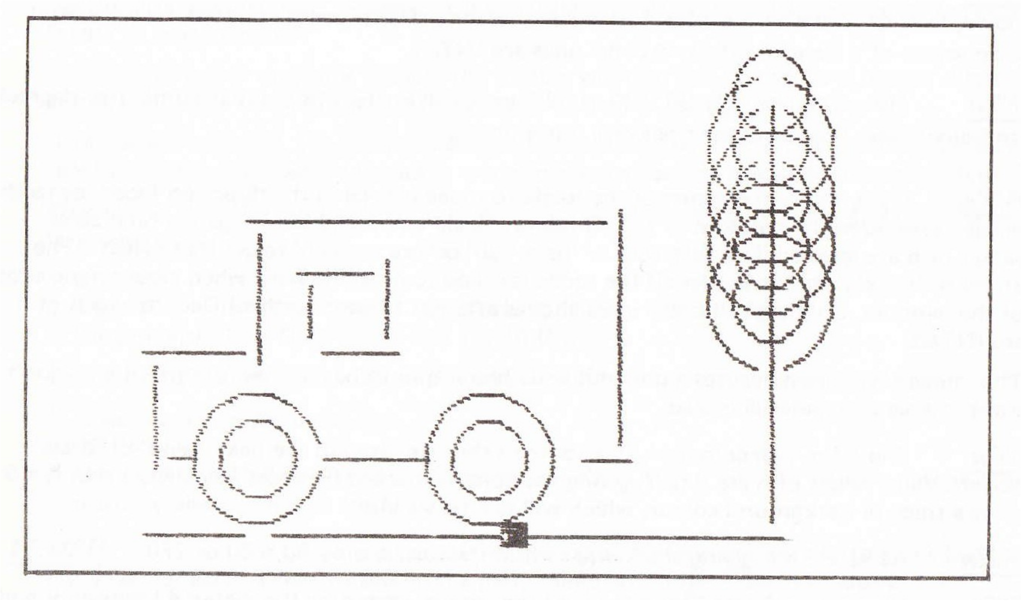
Procedures must appear after the 'endr' that terminates the main section and in numerical order.

Any allowed 'turtle' command may appear within a procedure except of course 'end' (and it is pointless to use 'endp' or 'stop'). However, while a procedure may call on another, or indeed itself, this normally leads to an infinite loop.

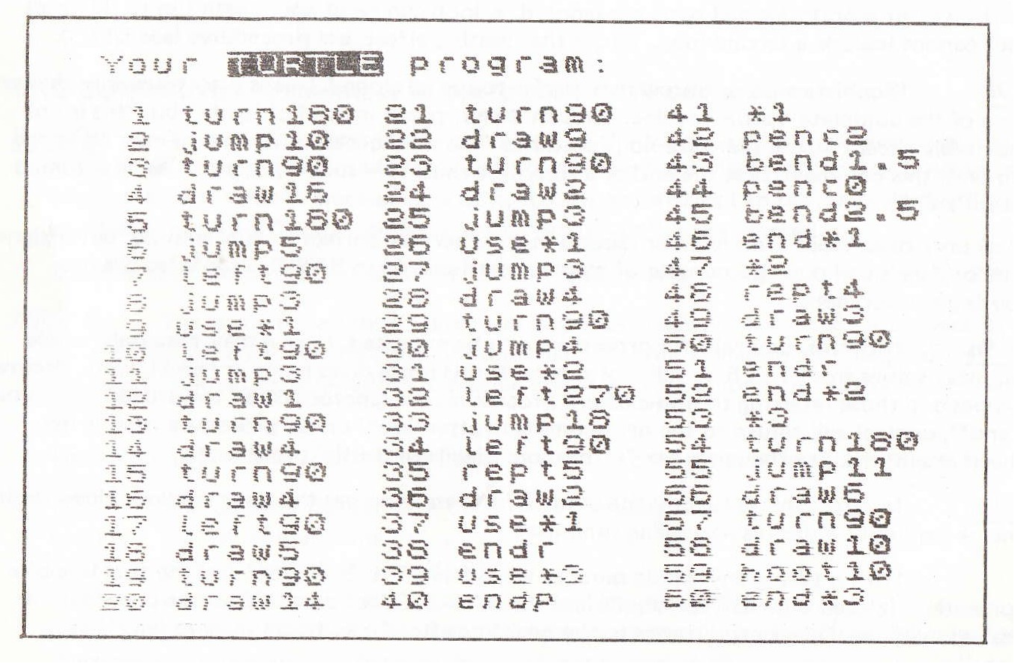
THE DEMONSTRATION ROUTINE

Note that we have used the word 'routine' throughout to make clear the difference between the 'turtle' program bought on cassette and the user's current set of 'turtle' instructions (commands). As purchased, the 'turtle' program includes within it a demonstration routine.

To see this in operation, press 7 at main menu stage. At the end of the' run, you get this screen display:



As you will see when you list with key 2 at main menu, the routine comprises the following lines:



```
   Your TURTLE program:
 1 turn180  21 turn90    41 *1
 2 jump10   22 draw9     42 penc2
 3 turn90   23 turn90    43 bend1.5
 4 draw16   24 draw2     44 penc0
 5 turn180  25 jump3     45 bend2.5
 6 jump5    26 use*1     46 end*1
 7 left90   27 jump3     47 *2
 8 jump3    28 draw4     48 rept4
 9 use*1    29 turn90    49 draw3
10 left90   30 jump4     50 turn90
11 jump3    31 use*2     51 endr
12 draw1    32 left270   52 end*2
13 turn90   33 jump18    53 *3
14 draw4    34 left90    54 turn180
15 turn90   35 rept5     55 jump11
16 draw4    36 draw2     56 draw6
17 left90   37 use*1     57 turn90
18 draw5    38 endr      58 draw10
19 turn90   39 use*3     59 hoot10
20 draw14   40 endp      60 end*3
```

Use this routine in turtle memory to see the effect of the other menu options, except key 3. Only use this EDIT option when you have a good idea of what is happening and how — and then use it to modify the various individual statements to explore to the full all the aspects of this version of turtle graphics.

Only then, really, can you develop your usage of the program with youngsters. When you do that, introduce them gently step by step to the commands and let them find out the syntax. At all times, remember the cardinal rule of turtling: WHEN IN DOUBT, PRETEND YOU'RE THE TURTLE!